CLAIMS

1. A network proxy server, comprising:

a network connection able to intercept content-object requests of clients from a server, and able to respond instead of said server to such; and

a plurality of content buffers for duplicating web content passing through from said server to any client, and for caching such web content to any subsequent clients;

wherein, multiple, moving-window buffers are included in the plurality of content buffers to service content requests of a server by various independent clients; and

wherein, whole requests for content-object from single clients can be serviced simultaneously from parts distributed across more than one such content buffer.

2. A system of delivering objects from servers to clients comprising:

receiving a first request for an content object from a first client;

allocating a first running buffer;

retrieving the content object as a datastream having a start point and inserting the datastream into the first buffer while delivering the same datastream to the first client;

when the first buffer is filled, deleting data from the start point of the datastream while continuing to insert retrieved data into the buffer, so that the buffer contains a moving window of the retrieved data;

receiving a second request for the content object from a client;

if the second request is received while the start point of the datastream is still in the first buffer, serving the content object directly from the first buffer; and

if the second request is received after the start point has been deleted from the first buffer, retrieving the portion of the content object that has been deleted from the first buffer, commencing from the start point, and delivering the same as a datastream while simultaneously delivering a different part of the content object from the first buffer.

3. The system of claim 2, further comprising, if the second request is received after the start point of the datastream has been deleted from the first buffer:

allocating a second running buffer and inserting the datastream representing the portion of the content object not in the first running buffer into the second running buffer while delivering the same datastream.

4. The system of claim 3 further comprising for a third request for the content object received after the second running buffer has been allocated;

checking whether the start point is cached in an existing running buffer;

if the start point is cached in an existing running buffer, serving the content object as a datastream from each of the running buffers simultaneously;

if the start point is not cached in an existing running buffer,

allocating a third running buffer;

retrieving the portion of the content object not in an existing running buffer as a datastream and inserting the datastream into the third running buffer while delivering the same datastream and simultaneously delivering a different part of the content object from other existing running buffers.

5. The system of claim 2, wherein the first buffer or another buffer has a size that is determined as a proportion of an advertised length of the content object.

6. The system of claim 2, further comprising:

modifying the size of the first buffer or another buffer in response to an analysis of frequency of requests for the content object, in order to optimize allocation of memory.

7. The system of claim 2, further comprising, prior to allocating the first buffer or another buffer, applying a replacement algorithm to reclaim buffers from less frequently requested objects.

8. The system of claim 2, wherein the content object has a time length L and each buffer has a start time S_i, an end time E_i and a running distance D_i, wherein the running distance D_i for each buffer after the first buffer equals: $D_i = S_i - S_{i-1}$, and wherein the end time E_i for each buffer after the first buffer is, $E_i = \min(S_{latest} + D_i, S_i + L)$, where, S_{latest} is the start time of the most recent buffer allocated.

9. Computer data storage media having stored thereon software performing the following functions:

receiving a first request for an content object;

allocating a first running buffer;

retrieving the content object as a datastream having a start point and inserting the datastream into the first buffer while delivering the same datastream;

when the first buffer is filled, deleting data from the start point of the datastream while continuing to insert retrieved data into the buffer, so that the buffer contains a moving window of the retrieved data;

receiving a second request for the content object;

if the second request is received while the start point of the datastream is in the first buffer, serving the content object directly from the first buffer;

if the second request is received after the start point has been deleted from the first buffer:

retrieving the portion of the content object that has been deleted from the first buffer, commencing from the start point, and delivering the same as a datastream while simultaneously delivering a different part of the content object as a datastream from the first buffer.

10. The computer data storage media of claim 9, wherein the software performs the following further functions:

if the second request is received after the start point of the datastream has been deleted from the first buffer, allocating a second running buffer and inserting the datastream representing the portion of the content object not in the first running buffer into the second running buffer while delivering the same datastream.

11. The computer data storage media of claim 9, wherein the software performs the following further functions:

receiving a third request for the content object after the second running buffer has been allocated;

checking whether the start point is cached in an existing running buffer;

if the start point is cached in an existing running buffer, serving the content object as a datastream from each of the running buffers simultaneously;

if the start point is not cached in an existing running buffer:

allocating a third running buffer;

retrieving the portion of the content object not in an existing running buffer as a datastream and inserting the datastream into the third running buffer while delivering the

same datastream and simultaneously delivering a different part of the content object as a datastream from other existing running buffers.

12. The computer data storage media of claim 9, wherein the software performs the following further functions:

determining the advertised length of the content object;

setting the size of the first buffer or another buffer as a proportion of an advertised length of the content object.

13. The computer data storage media of claim 9, wherein:

analyzing frequency of requests for the content object; and

modifying the size of the first buffer or another buffer in response to the analysis of the frequency of requests for the content object in order to optimize allocation of memory.

14. The computer data storage media of claim 9, wherein:

prior to allocating the first buffer or another buffer checking if memory is available;

if there is not enough memory available to allocate a buffer, applying a replacement algorithm to reclaim buffers from less frequently requested objects.

15. The computer data storage media of claim 9, wherein:

determining a time length L for the content object;

setting a start time S_i, an end time E_i and a running distance D_i for each buffer;

computing the running distance D_i for each buffer after the first buffer as, $D_i = S_i - S_{i-1}$;

computing the end time E_i for each buffer after the first buffer as, $E_i = \min(S_{latest} + D_i, S_i + L)$, where, S_{latest} is the start time of the most recent buffer allocated.